

(19)



JAPANESE PATENT OFFICE

## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **04191935 A**

(43) Date of publication of application: 10 . 07 . 92

(51) Int. Cl.

**G06F 9/46**  
**G06F 9/46**

(21) Application number: 02320994

(22) Date of filing: 27 . 11 . 90

(71) Applicant: **TOSHIBA CORP**
 (72) Inventor: **SUENAGA TSUKASA**  
**ISHIBASHI EIJI**  
**NOZAKI MASAHARU**

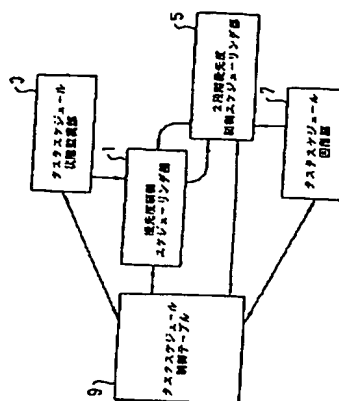
## (54) TASK SCHEDULING SYSTEM

(57) Abstract:

PURPOSE: To improve the performance of a system by alternately scheduling the task of high priority and the task of low priority.

CONSTITUTION: A task schedule state monitoring part 3 refers to a task schedule control table 9 every a constant period and shifts control from a priority control scheduling part 1 to a two-stage control scheduling part 5. The two-stage control scheduling part 5 classifies plural tasks into a task group with high priority and a task group with low priority and schedules the tasks of the low priority group in order by a low priority group pointer 17 of the task schedule control table 9. Then, a task schedule recovering part 7 shifts the control from the two-stage priority control scheduling part 5 to the priority control scheduling part 1 when the task finishes scheduling all the tasks of a CPU assignment queuing state. Thus, the performance of a system is improved.

COPYRIGHT: (C)1992,JPO&amp;Japio



⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

平4-191935

⑬ Int.Cl.

G 06 F 9/46

識別記号

3 4 0

B

庁内整理番号

8120-5B

3 2 0

D

8120-5B

3 4 0

E

8120-5B

⑭ 公開 平成4年(1992)7月10日

審査請求 未請求 請求項の数 1 (全6頁)

⑮ 発明の名称 タスクスケジューリング方式

⑯ 特 願 平2-320994

⑰ 出 願 平2(1990)11月27日

⑱ 発 明 者	末 永	司	東京都府中市東芝町1番地	株式会社東芝府中工場内
⑱ 発 明 者	石 橋	英 次	東京都府中市東芝町1番地	株式会社東芝府中工場内
⑱ 発 明 者	野 崎	正 治	東京都府中市東芝町1番地	株式会社東芝府中工場内
⑲ 出 願 人	株 式 会 社 東 芝		神奈川県川崎市幸区堀川町72番地	
⑳ 代 理 人	弁理士 三好 秀和		外1名	

明 細 書

1. 発明の名称

タスクスケジューリング方式

2. 特許請求の範囲

複数のタスクを備えて当該複数のタスクがCPUを時分割に供用するタスクスケジューリング方式において、

前記複数のタスクのうちCPU割当待ち状態で最も優先度の低いタスクの待ち時間を監視するスケジューリング監視手段と、

前記スケジューリング監視手段のCPU割当待ち状態のタスクの優先度の高いタスク群と優先度の低いタスク群とに分類して当該優先度の高いタスクと優先度の低いタスクとを交互にスケジューリングする2段階優先度制御手段と、

前記スケジューリング監視手段による待ち時間が指定値を超えると前記2段階優先度制御手段に制御を移行する2段階優先度制御移行手段と、

この2段階優先度制御移行手段により優先度の低いタスク群のうち優先度の最も低いタスクがス

ケジュールされると前記2段階優先度制御手段より優先度の高いタスクからスケジューリングする手段と、

を備えたことを特徴とするタスクスケジューリング方式。

3. 発明の詳細な説明

【発明の目的】

(産業上の利用分野)

本発明は、複数のタスクを備えたマルチタスクにおける当該タスクがCPUを時分割に供用するタスクスケジューリングにおいて、特に、優先度の高いタスクがCPUを占有し続けていても優先度の低いタスクがCPUを占有できることにより、システムのスループットを向上するタスクスケジューリング方式に関する。

(従来の技術)

従来のタスクスケジューリング方式は、タスクがCPU割当待ち状態(以下、レディ状態という。)であるとき優先度の高いタスクから順次、待ち行列に登録するタスクレディキューを備えて、

for a

タスクディスパッチング要求があるとタスクスケジューラは、当該タスクレディキューの先頭のタスクの優先度と現在CPUを占有中のタスクの優先度とを比較する。比較により優先度の高いタスクからスケジュールする。上記優先度の高いタスクをCPUに割当てたため、従来のタスクスケジューリング方式では、優先度の低いタスクをCPUに割当てるのが容易ではなかった。

(発明が解決しようとする課題)

ところで、従来のタスクスケジューリング方式は、優先度の高いタスクが、例えばプログラムのループ等によりCPUを占有して続けると優先度の低いタスクにCPUが割当てられなくなり、端末のレスポンスの低下およびシステム全体のスループットの低下を招来する問題があった。

本発明は、上記に鑑みてなされたものであり、その目的は、優先度の高いタスクがCPUを占有し続けている場合でも、優先度の低いタスクをCPUに割当てることにより、システムのスループットを向上させて、システムの性能を向上するタスク

スケジューリング方式を提供することにある。

[発明の構成]

(課題を解決するための手段)

上記目的を達成するため、本発明は、複数のタスクを備えて当該複数のタスクのうちCPU割当待ち状態で最も優先度の低いタスクの待ち時間を監視するスケジューリング監視手段と、

前記スケジューリング監視手段のCPU割当待ち状態のタスクの優先度の高いタスク群と優先度の低いタスク群とに分類して当該優先度の高いタスクと優先度の低いタスクとを交互にスケジュールする2段階優先度制御手段と、

前記スケジューリング監視手段による待ち時間が指定値を超えると前記2段階優先度制御手段に制御を移行する2段階優先度制御移行手段と、

この2段階優先度制御移行手段により優先度の低いタスク群のうち優先度の最も低いタスクがスケジュールされると前記2段階優先度制御手段から優先度の高いタスクからスケジュールする手段と、

を備えたことを要旨とする。

(作用)

上記構成を備えたタスクスケジューリング方式においては、複数のタスクのうちCPU割当待ち状態で最も優先度の低いタスクの待ち時間を監視して、当該待ち時間が指定値を超えると当該CPU割当待ち状態のタスクを優先度の高いタスクと優先度の低いタスクに分類されているタスクを交互にスケジュールする制御に移行する。前記スケジュールにより優先度の低いタスク群のうち優先度の最も低いタスクがスケジュールされると、通常の、優先度の高いタスクからスケジュールする制御に移行するので、優先度の高いタスクがCPUを占有している場合でも優先度の低いタスクがCPUを占有できる。

(実施例)

以下、図面を用いて本発明の実施例を説明する。

第1図は本発明のタスクスケジューリング方式に係る一実施例の概略を示すブロック図である。

上記タスクスケジューリング方式は、優先度制御スケジューリング部1、タスクスケジュール状態監視部3、2段階優先度制御スケジューリング部5およびタスクスケジュール回復部7を備える。

上記優先度制御スケジューリング部1は、優先度の高いタスクからスケジュールするものであり、タスクスケジュール状態監視部3は定周期毎に後述するタスクスケジュール制御テーブル9を参照して優先度制御スケジューリング部1から2段階制御スケジューリング部5に制御を移行するものである。2段階優先度制御スケジューリング部5は、複数のタスクを優先度の高いタスク群と優先度の低いタスク群とに分類して、後述するタスクスケジュール制御テーブル9の低優先度グループポインタ17により低優先度グループのタスクを順次、スケジュールする。

タスクスケジュール回復部7は、タスクがCPU割当て待ち状態(以下、レディー状態という。)のタスクを全てスケジュールし終ると2段階優先制御スケジューリング部5から優先制御スケ

ジャーリング部1に制御を移行する。

タスクスケジュール制御テーブル9は、以下に示す各種のポインタ等を備えており、第2図を用いて詳細に説明する。

上記タスクスケジュール制御テーブル(以下、テーブルという。)9はタスクスケジュール状態フラグ11およびタスクスケジュール状態変更上限カウンタ13を備えている。上記タスクスケジュール状態フラグ11は優先度制御スケジューリング部1および2段階優先度制御スケジューリング部5の高優先度グループ、低優先度グループを識別するフラグであり、例えば、優先度制御の場合「0」、2段階優先度制御の高優先度グループの場合「1」、低優先度グループの場合「2」の識別をしている。タスクスケジュール状態変更上限カウンタ13は、タスクスケジュール状態監視部3により、レディ状態のタスク $T_1, T_2 \dots T_n$ のそれぞれに備えられているタスク制御ブロック内の未スケジュール状態保持カウンタ $C_1, C_2 \dots C_n$ のうち、最も優先度の低いタスク、例え

ば $T_n$ の未スケジュール状態保持カウンタ $C_n$ と比較される。比較により上記タスクスケジュール状態変更上限カウンタ13の値が大きいときタスクスケジュール状態監視部3により2段階優先度制御スケジューリング部5に制御が移行される。

また、テーブル9は、被状態監視ポインタ15、低優先度グループポインタ17およびレディタスク制御ポインタ19を有する。

上記被状態監視タスクポインタ15は、2段階優先度制御に制御を移行する原因となったタスクを示すポインタであり、当該タスクが2段階優先度制御の最後にスケジュールされるタスクである。

低優先度グループポインタ17は、2段階優先度制御の低優先度グループのタスクを示すポインタである。

レディタスク制御ポインタ19は、レディ状態のタスクを優先度順に管理するエントリであり、当該レディ状態のタスク7が全てキューイング(タスクレディキュー)されている。

なお、タスクスケジュール状態監視部3等は、

オペレーティングシステム(以下、OSという。)により管理されている。

次に本実施例の作用を第3図および第4図の処理フローチャートを用いて説明する。

まず、第3図はタスク状態監視の処理を示すフローチャートである。

タスクスケジュール状態監視部3は、一定周期毎に起動され、通常の優先度制御の場合レディ状態のタスクのうち最も優先度の低いタスク、タスクレディキューの最後のタスクの未スケジュール状態保持カウンタをカウントアップし、現在CPUを占有しているタスクより優先度が低ければステップ160に進む(ステップ100~110)。

上記未スケジュール状態保持カウンタの値がタスクスケジュール状態変更上限カウンタ13の値を超えるとタスクスケジュール状態監視部3は、優先度制御スケジューリング部1から2段階優先度制御スケジューリング部5に制御を移行する。一方、未スケジュール状態保持カウンタの値がタスクスケジュール状態変更上限カウンタ13の値

以下ならばステップ160に進む(ステップ120)。

2段階優先度制御に移行されるとタスクスケジュール状態監視部3は、テーブル9の被状態監視タスクポインタ15をタスクレディキューの最後のタスクに設定し、低優先度タスクグループポインタ17をタスクレディキューの先頭のタスクに設定する。また、タスクスケジュール状態監視部3は、タスクスケジュール状態のフラグ11を2段階優先制御を示す、例えば「1」にセットしてステップ160に進む(ステップ130~150)。

ステップ160に進むとタスクスケジュール状態監視部3は、実行待ち状態になり一定周期後に再び起動されてステップ100に戻る(ステップ160)。

次にタスクスケジューリング処理および2段階優先度制御から回復処理を第4図のフローチャートを用いて説明する。

まず、タスクスケジュール状態監視部3は、テ

ーブル9のタスクスケジューリング状態フラグ11を調べ、通常の優先度制御のとき、現在CPUを占有しているタスクとレディキューの先頭のタスクの優先度とを比較する。比較によりレディキューのタスクの優先度が高いとき優先度制御スケジューリング部11は、当該タスクをスケジューリングして、CPUの使用権を持つカレントタスクをレディキューにつないだ後にCPUの使用権を与えるタスクスケジューリングを行なう。一方、CPUを占有しているタスクの優先度が高いとき優先度制御スケジューリング部11は、CPUの使用権を有するタスクのタスクスケジューリングを行なう(ステップ200~240)。

上記テーブル9のタスクスケジュール状態フラグ11が2段階優先制御を示すときタスクスケジュール状態監視部3は、現在までCPUを占有していたタスクをタスクレディキューに登録する。登録後、タスクスケジュール状態監視部3は、タスクスケジュール状態フラグ11が高優先度グループを示すときステップ340に進み、低優先度

グループを示すときステップ270に進む(ステップ250~260)。

ステップ270に進むと2段階優先度制御スケジューリング部5は、低優先度グループのタスクをスケジューリングするとき当該タスクがタスクレディキューの最後のタスクか、即ち、レディ状態のタスクを全てスケジューリングしたか否かを調べる。上記タスクがタスクレディキューの最後のタスクでない場合、2段階優先度制御スケジューリング部5は、低優先度グループポインタ17をレディキューの次のタスクに移すことにより、高優先度グループと低優先度グループの調整を行ない次の低優先度グループのタスクの実行を保証する。また、2段階優先度制御スケジューリング部5は、次のスケジューリングを高優先度グループで行なうためにタスクスケジュール状態フラグ11を高優先度グループに変更する(ステップ270~300)。

一方、タスクレディキューの最後のタスクの場合、2段階優先度制御スケジューリング部5は、

被状態監視ポインタ15、低優先度グループポインタ17およびレディタスク制御ポインタ19をそれぞれ初期化して、タスクスケジュールを元の優先制御に制御を移行する(ステップ310~330)。

タスクスケジュール状態フラグ11が高優先度グループを示すステップ340に進むと2段階優先度制御スケジューリング部5は、高優先度の先頭のタスクをスケジューリングしてステップ300に進む(ステップ340)。

これにより、優先度の高いタスクがCPUを占有している場合でも、優先度の低いタスクがスケジューリングすることにより、システムのスループットを向上できる。

また、デバックの作業中のシステムのバックアップにも対応可能になる。

#### 〔発明の効果〕

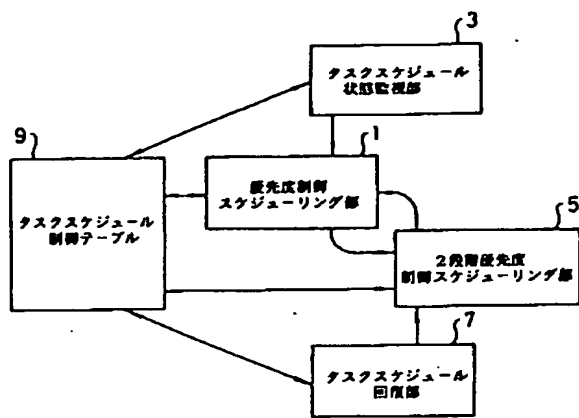
以上説明したように、本発明によれば、高優先度のタスクおよび低優先度のタスクを交互にスケジューリングするので、優先度の高いタスクがC

PUを占有し続けている場合でも、優先度の低いタスクをCPUに割当てることにより、システムのスループットを向上させて、システムの性能の向上を実現できる。

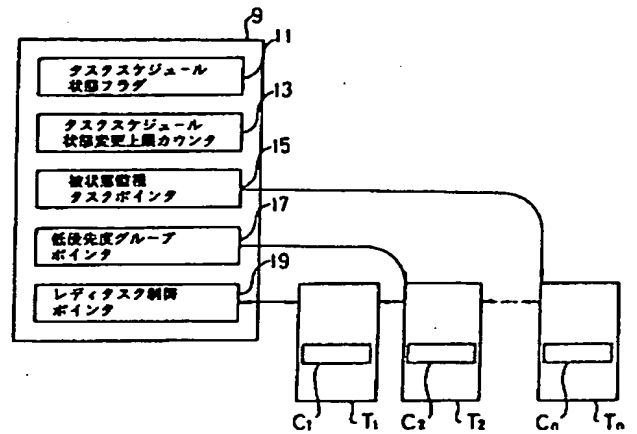
#### 4. 図面の簡単な説明

第1図は本発明のタスクスケジューリング方式に係る一実施例を示す概略図、第2図はタスクスケジューリング制御テーブルを示す図、第3図および第4図は本発明の動作を示すフローチャートである。

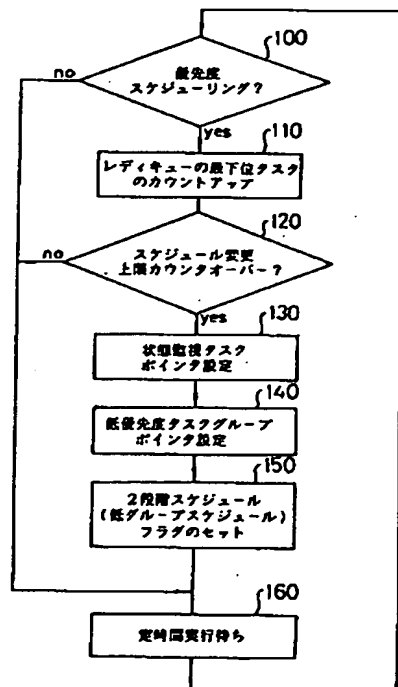
- 1…優先度制御スケジューリング部
- 3…タスクスケジュール状態監視部
- 5…2段階優先度制御スケジューリング部
- 7…タスクスケジュール回復部
- 9…タスクスケジュール制御テーブル
- 11…タスクスケジュール状態フラグ
- 13…タスクスケジュール状態変更上限カウンタ
- 17…低優先度グループポインタ



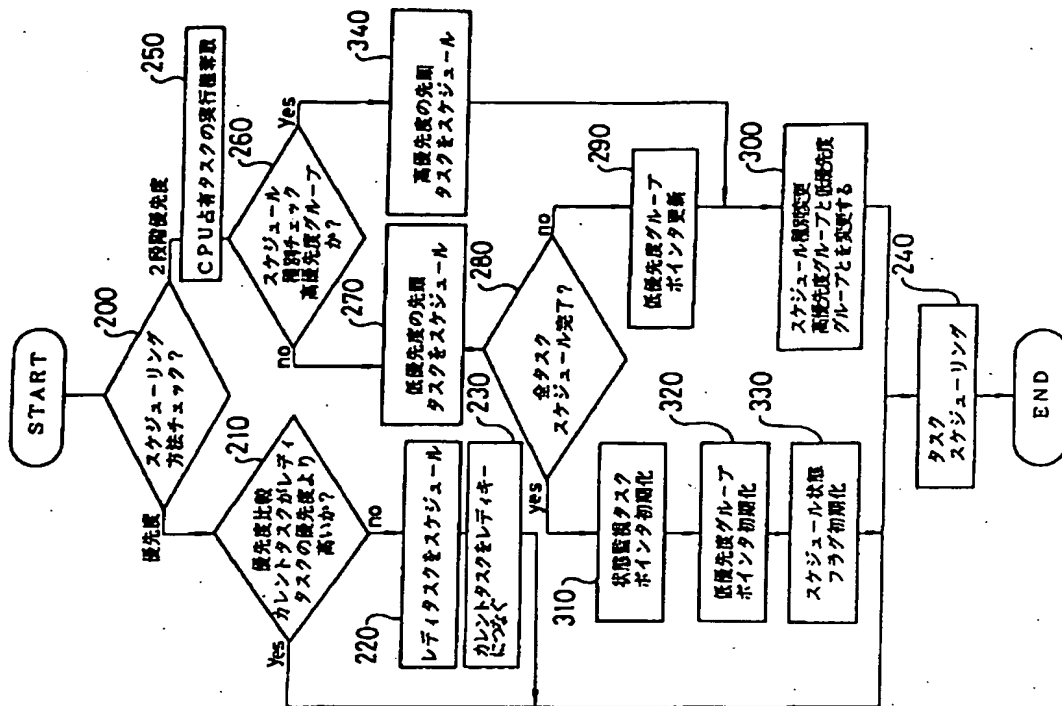
第 1 図



第 2 図



第 3 図



第 4 図